Axioms as they currently stand:

```
class EuclideanPlane where
  Pt : Type
  Line : Type
  Circ : Type
  Pt_on_Line : Pt → Line → Prop
  Line_of_Pts : ∀ a b : Pt, ∃ L : Line, (Pt_on_Line a L) ∧ (Pt_on_Line b L)
  Pt_on_Circ : Pt → Circ → Prop
  Center_of_Circ : Pt → Circ → Prop
  Circ_of_Pts : ∀ a b : Pt, ∃ α : Circ, (Center_of_Circ a α) ∧ (Pt_on_Circ b
α)
  dist : Pt → Pt → ℝ
  dist_symm : ∀ (a b : Pt), dist a b = dist b a
  dist_nonneg : ∀ (a b : Pt), 0 ≤ dist a b
  dist_pos_def : ∀ (a b : Pt), dist a b = 0 ↔ a = b
  Circs_inter : Circ → Circ → Prop
  Pts_of_Circs_inter : ∀ α β : Circ, Circs_inter α β → ∃ a b : Pt,
    Pt_on_Circ a α ∧ Pt_on_Circ b α ∧ Pt_on_Circ a β ∧ Pt_on_Circ b β ∧ a ≠
b
  Pt_in_Circ : Pt → Circ → Prop
  Circs_inter_iff : ∀ (α β : Circ), Circs_inter α β ↔
    (∃ (a b : Pt), Pt_on_Circ a α ∧ Pt_in_Circ a β ∧
      Pt_on_Circ b β ∧ Pt_in_Circ b α)
  Pt_on_Circ_iff : ∀ (a b c : Pt) (α : Circ), Center_of_Circ a α →
    Pt_on_Circ b α → (Pt_on_Circ c α ↔ dist a c = dist a b)
  Pt_in_Circ_iff : ∀ (a b c : Pt) (α : Circ), Center_of_Circ a α →
    Pt_on_Circ b α → (Pt_in_Circ c α ↔ dist a c < dist a b)
```

Homework was to work on:

```
lemma in_Circ_of_center (a : Pt) (α : Circ) (a_cent_α : Center_of_Circ a α)
:
          Pt_in_Circ a α := by
    sorry -- HOMEWORK
```

It seems that we want to be able to describe a point on the circle `α` and yet there's no axiom that tells us that we can actually produce a point on a circle! Let's add that axiom.

```
    Pts_of_Circ : ∀ (α : Circ), ∃ (b c : Pt), Pt_on_Circ b α ∧ Pt_on_Circ c
α ∧ b ≠ c
```

Given this new axiom, we can start the proof of `in_Circ_of_center` with:

```
    have := Pts_of_Circ α
    obtain ⟨b, c, b_on_α, c_on_α, b_ne_c⟩ := this
```

Now what? To show that `a` is inside `α`, we'll need to use `Pt_in_Circ_iff`. Start by `hav`ing it: When we write

```
have := (Pt_in_Circ_iff a b a α a_cent_α b_on_α)
```

we produce a new hypothesis (by default, called `this`):

```
this : Pt_in_Circ a α ↔ dist a a < dist a b
```

The "if and only if" `↔` is secretly a pair of statements; the first statement is:

```
Pt_in_Circ a α → dist a a < dist a b
```

and the second statement is:

```
dist a a < dist a b → Pt_in_Circ a α
```

Which one is useful for us? The second one, since it's the one that concludes that `a` is inside `α`. So we put parentheses and `.2` to say, give me the second statement.

```
    have := (Pt_in_Circ_iff a b a α a_cent_α b_on_α).2
```

Recall that if we have a hypothesis `H : P → Q` and a goal `⊢ Q`, then if we `apply H`, the goal will change to `⊢ P`. In our case, we'll type `apply this`.
We realized that we could use another helper lemma, which proves that `dist a a = 0`.

```
-- An easier helper lemma:
lemma dist_self (a : Pt) : dist a a = 0 := by
    have := (dist_pos_def a a).2
    apply this
```

```
    -- now the goal is to prove `⊢ a = a` and that's solved by
    -- the reflexive property of the equal sign, called `rfl`
    rfl
```

Again, if your goal is to prove that two things are equal, and they're literally the same thing, for example: `⊢ x ^ 2 - 7 x + 6 = x ^ 2 - 7 x + 6`, that is solved by: `rfl`.

Let's go back to our orginal (API = application programming interface = lemma); we added `have dist_aa : dist a a = 0 := dist_self a`, and got the goal state:

```
**a** : Pt
**α** : Circ
**a_cent_α** : Center_of_Circ a α
**b c** : Pt
**b_on_α** : Pt_on_Circ b α
**c_on_α** : Pt_on_Circ c α
**b_ne_c** : b ≠ c
**this** : dist a a < dist a b → Pt_in_Circ a α
**dist_aa** : dist a a = 0
**⊢** dist a a < dist a b
```

Here's the issue: We know that `dist a a = 0`, and that `0 ≤ dist a b`, and we want to show that `dist a b` is *strictly* positive. But where are we supposed to get that from? One idea: add another axiom that somehow represents the fact that circles have positive radius.
Claim: That can already be proved from the axioms that we have! Why?

Homework: prove this helper lemma:

```
-- Another helper lemma: the "radius" is always strictly positive
lemma radius_pos (a b : Pt) (α : Circ)
(a_cent_of_α : Center_of_Circ a α)
(b_on_α : Pt_on_Circ b α) :
0 < dist a b := by
-- idea:
-- by `dist_nonneg`, either `0 = dist a b` or `0 < dist a b`
-- if the second case, then we're done. So we just want to rule
-- out the first case
-- Now we're arguing by contradiction (we want to prove `false`!)
-- Using `Pts_of_Circ`, we can `obtain` two new points, say, `e` and
-- `f` that are distinct and both on `α`.
```

```
-- If `dist a b = 0` then `a = b` (by `dist_pos_def`).
-- But `Pt_on_Circ_iff` says that radii are well-defined
-- that is, `dist a e = dist a b = 0 = dist b e` So
-- `b = e` (again by `dist_pos_def`). The same argument proves
-- `b = f`, which by transitivity proves `e = f`.
-- Which is a contradiction!
-- Hint: `e ≠ f` by defintion means: `e = f → false`.
sorry
```

Prove in natural language (extra credit: prove formally!)